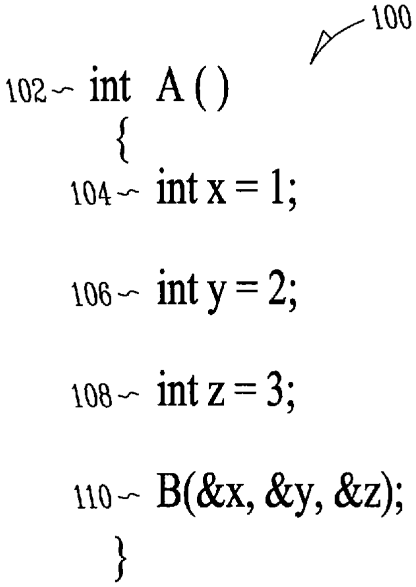


```
int  A ()
  {
    int x = 1;

    int y = 2;

    int z = 3;

    B(&x, &y, &z);
  }
```

FIG. 1

200

202	0x100	x
204	0x0FC	y
206	0x0F8	z
208	0x0F4	z
210	0x0F0	y
212	0x0EC	x
214	0x0E8	RTN
216	0x0E4	BP
218	0x0E0	

FIG. 2

(PRIOR ART)

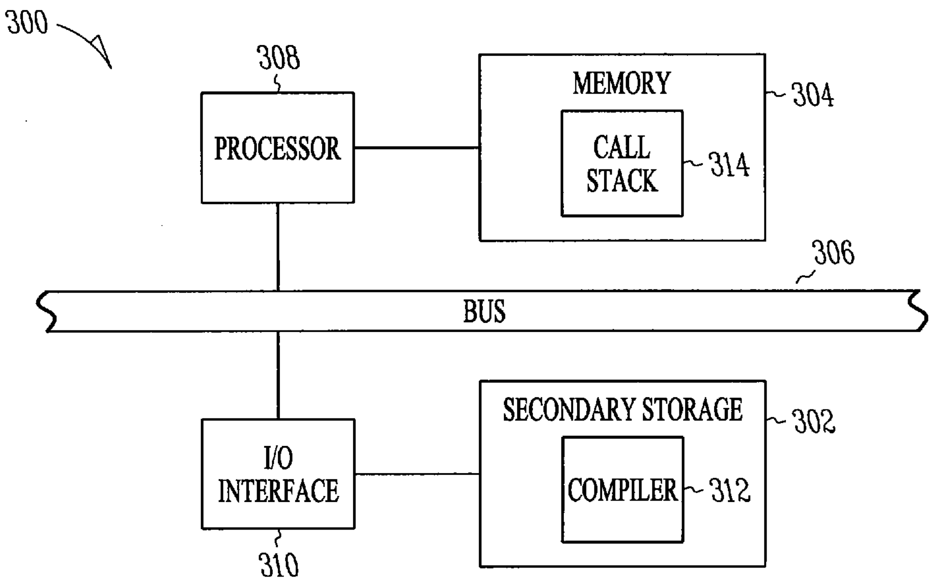
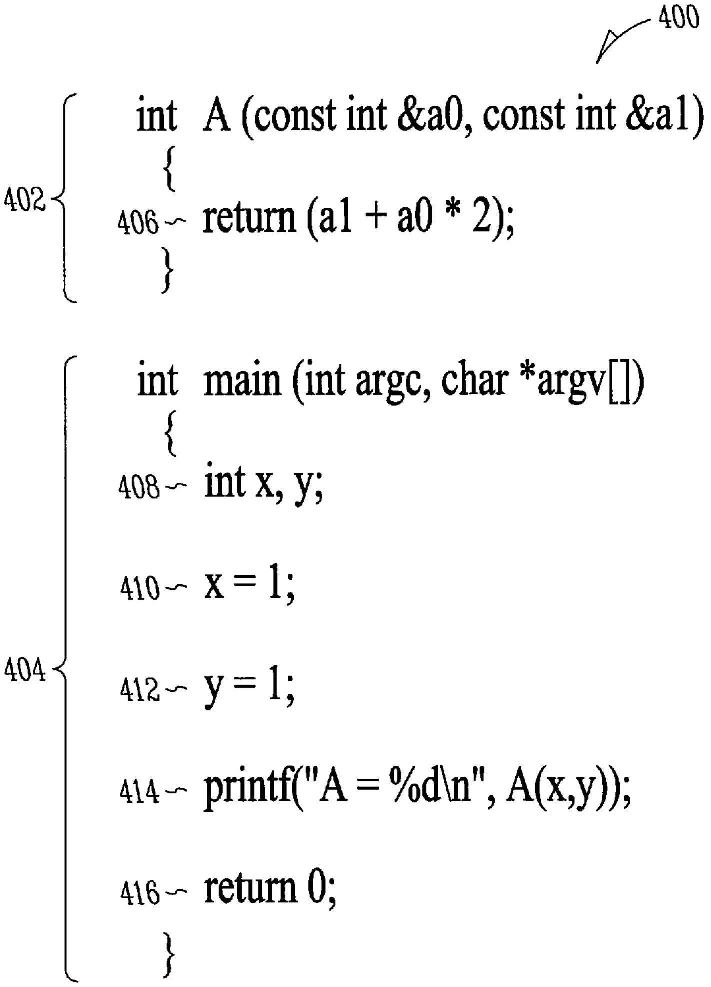
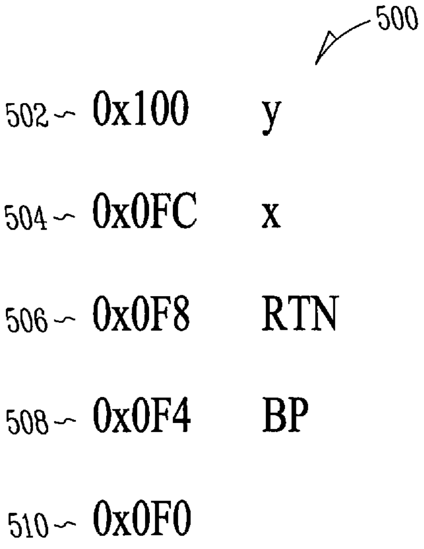

300
308
PROCESSOR
MEMORY
CALL STACK
314
304
306
BUS
SECONDARY STORAGE
302
COMPILER
312
I/O INTERFACE
310

FIG. 3



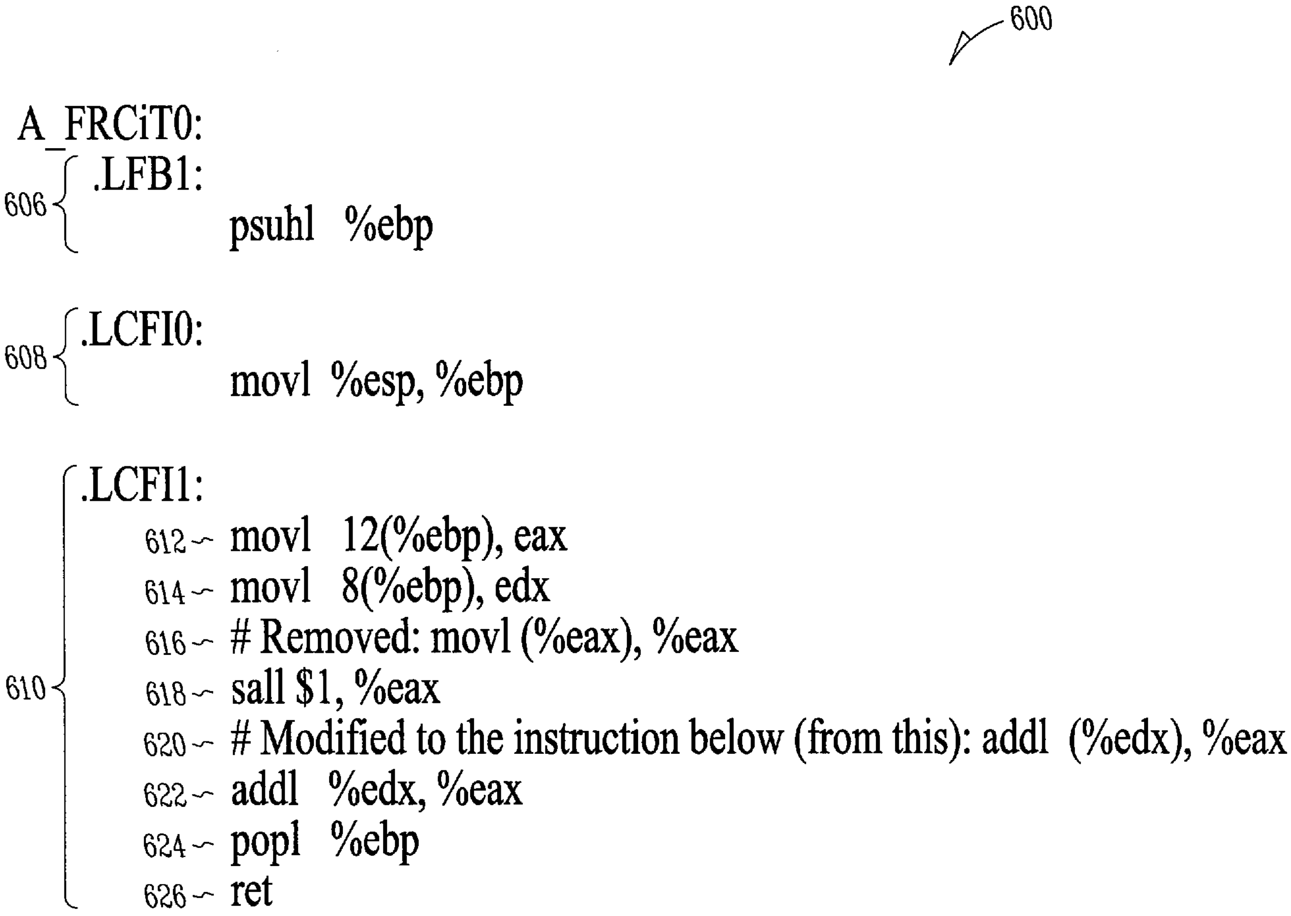
```
int  A (const int &a0, const int &a1)
  {
  return (a1 + a0 * 2);
  }

int  main (int argc, char *argv[])
  {
  int x, y;

  x = 1;

  y = 1;

  printf("A = %d\n", A(x,y));

  return 0;
  }
```

FIG. 4



```
0x100   y

0x0FC   x

0x0F8   RTN

0x0F4   BP

0x0F0
```

FIG. 5



```
A_FRCiT0:
  .LFB1:
          psuhl  %ebp

 .LCFI0:
          movl  %esp, %ebp

 .LCFI1:
          movl   12(%ebp), eax
          movl   8(%ebp), edx
          # Removed: movl (%eax), %eax
          sall $1, %eax
          # Modified to the instruction below (from this): addl  (%edx), %eax
          addl   %edx, %eax
          popl   %ebp
          ret
```

FIG. 6A

640

```
main:
642 { .LFB2:
                psuhl  %ebp

644 { .LCFI2:
                movl  %esp, %ebp

646 { .LCFI3:
                subl   $8, %ebp

648 { .LCFI4:
         660~ movl   $1, -4(%ebp)
         662~ movl   $2, -8(%ebp)
         664~ # Removed: subl $8, %esp
         667~ # Removed: leal -8(%ebp), %eax
         668~ # Removed: pushl    %eax
         670~ # Removed: leal -4(%ebp), %eax
         672~ # Removed: psuhl    %eax

650 { .LCFI5:
         674~ call     A_FRCiT0
         676~ #Removed    addl    $8, %ebp
         678~ pushl    %eax
         680~ pushl    $.LC0
```

FIG. 6B

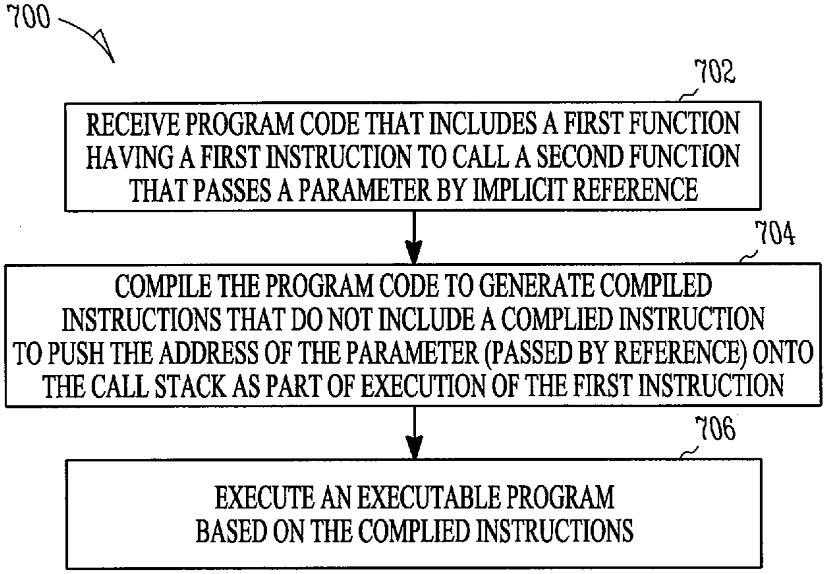

700
702
RECEIVE PROGRAM CODE THAT INCLUDES A FIRST FUNCTION HAVING A FIRST INSTRUCTION TO CALL A SECOND FUNCTION THAT PASSES A PARAMETER BY IMPLICIT REFERENCE
704
COMPILE THE PROGRAM CODE TO GENERATE COMPILED INSTRUCTIONS THAT DO NOT INCLUDE A COMPLIED INSTRUCTION TO PUSH THE ADDRESS OF THE PARAMETER (PASSED BY REFERENCE) ONTO THE CALL STACK AS PART OF EXECUTION OF THE FIRST INSTRUCTION
706
EXECUTE AN EXECUTABLE PROGRAM BASED ON THE COMPLIED INSTRUCTIONS

FIG. 7

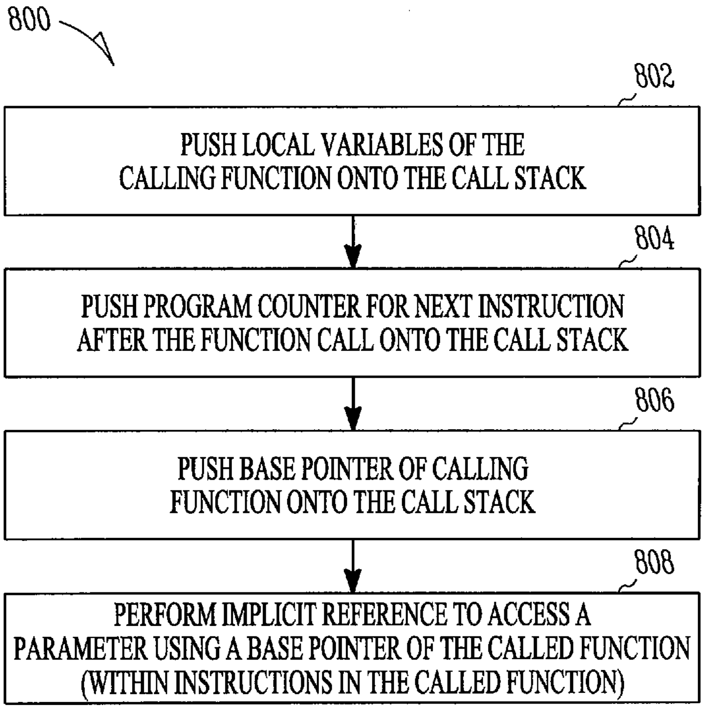

800
802
PUSH LOCAL VARIABLES OF THE CALLING FUNCTION ONTO THE CALL STACK
804
PUSH PROGRAM COUNTER FOR NEXT INSTRUCTION AFTER THE FUNCTION CALL ONTO THE CALL STACK
806
PUSH BASE POINTER OF CALLING FUNCTION ONTO THE CALL STACK
808
PERFORM IMPLICIT REFERENCE TO ACCESS A PARAMETER USING A BASE POINTER OF THE CALLED FUNCTION (WITHIN INSTRUCTIONS IN THE CALLED FUNCTION)

FIG. 8

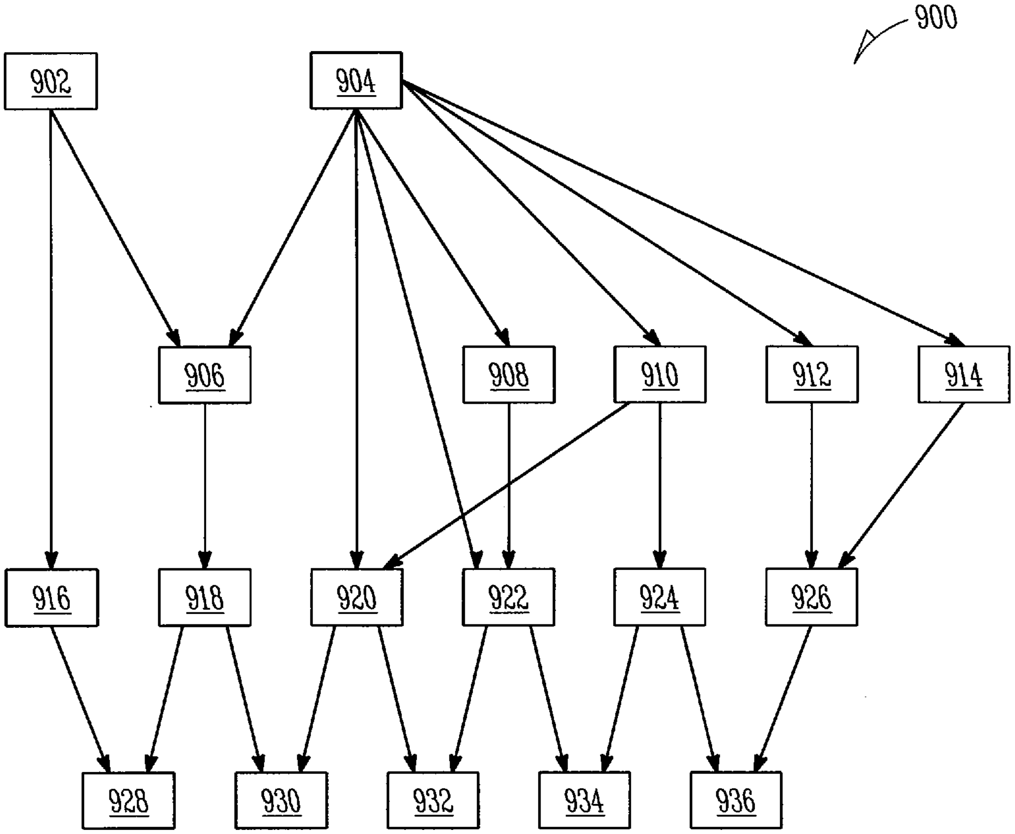


FIG. 9